Andrew McPherson  
CS-340  
6/19/2022

**7-2 Project Two Submission:  
&  
README**

**About this Project**

The required functionality of this project was to create a viewable data table from a database of animals that may be used for a company Grazioso Salvare. The intended functionality is to not only show this data in a meaningful way, but to also add filtering options to the data table to help this company search through for specific dog breeds that they would need for various rescue purposes.

These filtering options were also required to be displayed in other meaningful ways such as geolocation chart that would update based on filtering, and a Pie chart that would show the filtered data more clearly based on breeds.

To demonstrate this functionality, below is attached a screenshot showing the data table as well as these two additional charts  
**Note**: I mention this at the bottom of this paper, but the remainder of screenshots that include the different filter buttons in action do **not** show the name/image each time. They *are* always there but only shown here for saving space on the paper.

Table

Description automatically generated

Map

Description automatically generatedHere you can see the dashboard upon first booting up and loading (in two parts). It starts in “reset” which shows the default data table for all items.

**Tools Used and Rationale:**

**MongoDB-**

MongoDB was the choice of tool used for the database component because MongoDb can easily import a large CSV document into a database, and then we are able to use Pymongo to create python code for easier manipulation of said database. Also, due to MongoDB requiring a bit more involved code to modify documents in a database collection having the ability to use Python and Pymongo for our CRUD operations makes it generally easier to use, and modular.

**Dash-**

Dash is the tool that was used to build the visual elements of the dashboard and the database, it is a Python framework that mostly works with HTML, so it allows easy ways to visualize data such as this.

**Reproduction of Project:**

Since this project works with a MongoDB collection, and through a Python Module, on top of the Dash framework, to get to a point of recreating this project you would need to create a database called AAC, which from there you will create a user that has read/write access to this database.  
You would then need to import the data from the aac\_shelter\_outcomes.csv file.  
Once you have all of these components in place, reproduction of this project could be done by using the python module created for this database. The python module includes methods for CRUD functionality, so there is a method for Create, Read, Update, and Delete. I also had included a method to “Read All” which was necessary for showing the entire database on the data table. This module also of course needs to have functionality to connect to the Mongo database properly, which is one of the requirements for the read/write user that was created for the purpose of this project.  
Once all this is in place you would simply need to use the dash framework to create the data table by accessing the mongo database through this python module.

**Steps taken for project completion:**

Due to using provided code, I will be simply addressing the steps I took from there,  
But otherwise you would need to start this project with loading all the correct imports.  
  
I started the project by correctly importing my project.  
Logo

Description automatically generated with low confidence  
Form there, I set up variables for the user/password that I will be using for connecting to the database, and then creating an object instantiation of the AnimalShelter class from the module with that username and password like so  
Text

Description automatically generated with medium confidence

Now that the class object is instantiated and we are connected to the database, we will start by performing the readAll function to populate the dashboard to be used.

I set up the layout for the dashboard, also added the image, and added radio button functionality to the top of the data table which will be used for filtering said data table  
Text

Description automatically generated

Text

Description automatically generated

The next thing I had to do was to create the proper callback to add the functionality of those radio buttons from before, so here is an example of one of these filtering options I had to create.

Text

Description automatically generatedText

Description automatically generatedThe same was also done for a simple pie chart and a geolocation chart. Below is the pie chart code example

From there the data table was showing correctly. Below are the remainder screenshots that demonstrate the radio button functionality:

Once the radio button for “Water Rescue” is selected, this is the updated datable and charts

**Note:** For these additional screenshots below I did **not** include the image or name header that is in between the data table and the charts, but they **were** present every time.  
Once you see it one time (as you would have above at the top) I did not think it necessary to waste space on this paper by showing them for 3 other buttons.

Map

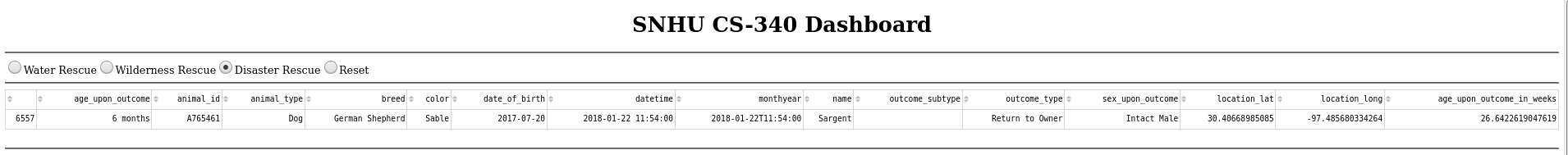
Description automatically generatedTable

Description automatically generated with medium confidenceMap

Description automatically generatedTable

Description automatically generatedThen, once again when clicking on “Wilderness Rescue” this is the following data table and charts.

Map

Description automatically generatedAnd then for “Disaster Rescue” is selected it shows the following data table (only one entry) and the charts.

**Challenges in the project:**

The use of the HTML-like format of Dash was something new for me to work on, and the hardest challenge of this all was trying to understand proper indenting and format structure because it does not take much for a line to be out of place and cause the whole output to stop working and displaying anything, but neither will you receive an error telling you what or where the problem was.  
To overcome these challenges, it simply involved a lot of research into examples of how this code is to be created.  
Other challenges were trying to learn how to add functionality to the radio buttons, which is an area I struggled with to begin with as getting the radio buttons to work was also an issue that took more time and research to learn how to use.

I also struggled adding further and more complicated functionality to the Geolocation chart, such as updating the marker location based on the filter settings.  
The information displayed from the marker was correct, but not the position.  
Majority of these challenges are also simply just “testing”, test the code, see what runs. If I get an error, I can see where it is and try to fix it. Challenges where the output breaks and I see nothing while also receiving no errors are near impossible to fix without hitting undo a lot until it is working again, so there are some areas that need improvement.

**Known Issues:**

As mentioned above, adding further functionality into the Geolocation chart was not done, so currently the position marker does not change location as it is hard coded in, but the data does change that is being represented by said marker. It will update the name and breed of the first dog of the table after using the radio button filters, it reacts dynamically, but I was unable to learn how to add in proper location markers.